09/308300
Rec'd PCT/PTO 17 MAY 1999
PCT/US96/18477

COMPUTER TUTORIAL AND EXECUTION INTERFACE OPERABLE OVER NETWORK CONNECTION

Technical Field

The present invention relates generally to computers, and, specifically, to an execution system operable to perform tutorial operations, expert system operations, document preparation operations, financial analysis preparation, and various other functions in conjunction with various computer application programs and also operate various peripheral components to perform operations without requiring a specific interface to each application program and/or peripheral component.

Background of the Invention

For many years, application programs, on-line services, and other computer application software have been available for use with digital computers. Application programs performed word processing functions, numeric functions, data-base functions, accounting functions, inventory control functions, and a wide variety of other functions. Application programs served not only to increase the efficiency of their user but to increase the user's accuracy as well. On-line services allowed a user to access large databases of information that could be downloaded and used as desired.

Application programs may be used in document preparation, financial analysis preparation, communication preparation, monitoring of the operation of systems and components through connected peripheral devices, and numerous other functions. Application programs, however, generally operate in isolation, not having the ability to interface with other programs, except those of a common vendor, at best. When application programs do work together, it is typically through the common formatting and interchange

of data. Thus, no content is typically provided by the interaction of application programs. Users of the application programs operate in isolated environments with the programs.

When first operating an application program or on-line service, a significant amount of time is required to educate the user in the use of the computer application software or information services. Not until the user is sufficiently trained in the use of the computer application software or service may substantial benefits be derived from the program or service.

The earliest approaches to training users were made by the providers of computer application software. The first training tools involved written instruction books that were included with the programs. These books described the functions available in the software, how to implement the functions, and the limitations of the functions. The earliest training books were written in a highly technical manner that prevented the average user from gaining a thorough understanding of the program. Resultantly, a large industry grew around providing written training materials for training users to use the programs. Over time, with increased competition, the written materials became easier to read and understand, providing more thoughtful approaches to educating the user.

While the software industry developed, some persons and organizations recognized the shortcomings of the written book type instruction and stepped in to provide classroom and interpersonal instruction. Classroom type instruction targeted specific software that was popular enough to justify the large capital expenses associated with this type of training. Levels of instruction varied from lectures given in large auditoriums all the way down to one-on-one training sessions. While this type of instruction proved to be quite successful due to its human aspect, it was very expensive and generally required that the new user leave his or her place of employment to attend. Further, because the user generally was not provided with hands-on training, and even if he or she was, the training was not performed

on his or her own machine. Because of differences in machines and environments used in the training classes, the user could not always transfer the knowledge he or she had obtained to his or her own computer.

Over time, vendors of the software and others in the industry recognized the value of training the user on his or her own machine while inside the software itself. Thus, on-line tutorials were developed. On-line tutorials typically combined a written description of a particular function of the software and instruction in specific commands that would allow a user to perform the function. These on-line tutorials typically allowed a user to perform a few instructions at a time as directed by the tutorial with the instructions being monitored to ensure correctness of operation. While these on-line tutorials provided the benefit of learning while doing, they were typically difficult to follow and did not provide adequate explanation. Part of the problem related to inadequate written explanation that carried over from the user's manual was that such information was conveyed to the user only in a written format displayed on the screen. Further, because they were specific to the particular application program, they did not provide a familiar reference frame for the user, and the user first had to learn to use the on-line tutorial in the particular program.

Thus, attempts were made to combine the benefits of classroom training with the benefits of hands-on training on the user's own machine. A few vendors recorded classroom training programs on video cassettes so that a user could play the topics at his or her own speed on a nearby television while simultaneously working on the computer. Thus, a user could combine the benefits of working on his or her own machine while also obtaining the benefits of being in a classroom. Unfortunately, there was no interplay between the video being viewed and the user's commands issued to the computer application software. While this system allowed the user to play the video in his of her office, it did not provide the interactive benefits available from other techniques. Thus, the system did not reinforce the

commands described in the video and required the simultaneous operation of two separate machines.

One visual teaching aid, sold under the tradename LOTUS SCREENCAM, displays images on a computer screen that are identical to those displayed within an application program. However, even though the teaching aid displays images that a user would encounter during use of the program, the teaching aid merely functions like a video player. The teaching aid merely displays to the user a proper sequence of keystrokes and/or mouse movements that would be required to execute specific functions and does not provide interaction between the user and the actual application program.

Even when a user becomes proficient in the use of an application program, the user must provide all content to documents produced by the application program. Thus, the training performed and expertise gained typically has no relationship to particular job tasks to be performed. Some application programs included wizard applications or functions that aided the user in performing certain tasks within the application programs. These wizard applications are a form of internal control or operation of the application programs. But the variety of tasks capable of being executed by such internal controls is typically limited by several factors, including the fact that the variety of tasks is buried into the internal structure of the application itself, and thus not readily accessible, and the internal level of complexity added by these internal controls or operations prevent vendors from covering a broad range of capabilities.

Another problem faced by currently-available systems relates to the control of peripheral devices connected to computer systems. These devices may comprise interface circuitry allowing the computer system to function with equipment, such as process controllers, manufacturing equipment, and other equipment that may be computer controlled. Software for controlling these devices was typically highly specific to the type

of equipment being controlled, could not interact with other computer programs, and provided little ability for users to vary the operation of connected equipment.

Thus, there lies a need for a multipurpose execution system that interfaces between application programs, local data bases, remote data bases, expert systems, and peripheral devices to perform various functions including training, control, and assistance to a user.

Summary Of The Invention

The invention herein is provided to overcome the above-described limitations, and others, of the prior tutorials, application program interfaces and execution systems. The present invention, therefore, includes a user interface, a data retrieval interface, a data interpreter, a target application interface, and a peripheral device interface. The user interface transmits an instruction set to a user and receives instructions from the user based upon the instruction set. One of the received instructions selects a target application or peripheral device from a plurality of independently executable applications or peripheral interfaces.

The data retrieval interface retrieves data from storage based upon user input while the data interpreter receives the data and the instructions from the user and, based upon the target application and the data, transforms the data into actions. The target application interface receives at least some of the actions and selectively issues some of the actions to operate the target application. Finally, the peripheral interface receives at least some of the actions and selectively issues some of the actions to operate the peripheral device. Thus, the system of the present invention provides a multi-purpose generalized interface to any application program or peripheral device.

The present invention retrieves data and instruction sets from either a local data archive such as a CD ROM or magnetic disk or retrieves instructions and data from a remote

storage location over a data link. The instructions include generalized task sets and transformation components. The transformation components allow the system to transform the generalized task sets for particular application programs or peripheral devices. Thus, the system of the present invention has a wide range of applicability.

More particularly, one embodiment of the system provides a computer-based tutorial interface that provides a user with audio/visual training on specific functions of a computer program or computer services, provides actual samples of the implementation of the functions, assists the user in learning to perform the function within the computer application software, and requires the user to take an active approach in the training by performing actual instructions within the application software on the user's own machine.

Moreover, other aspects of the present invention will become apparent with further reference to the drawings and specification which follow.

Brief Description Of The Drawing

FIG. 1 is a block diagram illustrating a computer system that may be utilized in conjunction with the present invention;

FIG. 2 is a functional block diagram illustrating the functional relationship between elements of an embodiment of a system embodying the principles of the present invention;

FIG. 3 illustrates a sample display screen of an interface of a system or the present invention performing tutorial functions detailing a tutorial topic for a paint-type application program;

FIGs. 4a and 4b illustrate executions accomplished by the system of FIGs. 2 and 3 relating to particular selections made by a user of available functions;

FIG. 5 illustrates a further embodiment of the invention with an action flow of instructions and data among the elements of the system wherein the instructions are selectively interpreted and forwarded based upon a selected target application;

FIG. 6 is a block diagram of the system of the present invention illustrating the flow of instructions and feedback;

FIG. 7 is a flow diagram illustrating the various action steps involved in retrieving data from either a local or a remote storage location;

FIB. 8 is a flow diagram illustrating the various action steps involved in issuing instructions to the appropriate interface, that is, a target application or a peripheral device interface;

FIG. 9 is illustrates a conventional format of an instruction file including initialization instruction header and the sequence of elements of an elementary instructions; and

FIG. 10 is an example of a library field structure that can be available to the system.

Detailed Description of a Preferred Embodiment

In the following description, certain details are set forth to provide a complete understanding of the present invention. It will be apparent to one skilled in the art, however, that these specific details are not required in order to practice the present invention. Also, well known electrical structures and circuits are depicted in block diagram form so as not to obscure the present invention unnecessarily.

The descriptions which follow are presented in part in terms of operations within a computer. These descriptions and representations are the means used by those skilled in the software arts to most effectively convey the substance of their work to others skilled in the art.

An algorithm is here, and, generally, conceived to be a self-consistent sequence of steps leading to a desired result. These steps are those requiring physical manipulations of physical quantities. Usually, though not necessarily, these quantities take the form of electrical or magnetic signals capable of being stored, transferred, combined, compared, and otherwise manipulated. It proves convenient at times, principally for reasons of common usage, to refer to these signals as, values, symbols, characters, display data, terms, numbers, or the like. It should be borne in mind, however, that all of these and similar terms are to be associated with the appropriate physical quantities and are merely used here as convenient labels applied to these quantities.

Further, the manipulations performed are often referred to in terms, such as comparing, commonly associated with mental operations performed by a human operator. No such capability of a human operator is necessary, or desirable in most cases, in any of the operations described herein which form part of the present invention, since the operations are machine operations. Useful machines for performing the operations of the present invention include general purpose digital computers or other similar devices. The present invention relates to method steps and apparatus for operating a computer in processing electrical or other physical signals to generate other desired physical signals.

The present invention also relates to a system for performing these operations. This system may be specifically constructed for the required purposes, or it may comprise a general purpose computer as selectively activated or reconfigured by a computer program stored in the computer. The algorithms presented herein are not inherently related to any particular computer or other apparatus. In particular, various general purpose machines may be used with programs written in accordance with the teachings herein, or it may prove more convenient to construct more specialized apparatus to perform the required method steps.

The required structure for a variety of these machines will appear from the description below.

In the following description, several terms are used frequently, have specialized meanings in the present context, and are thus defined. The terms "environment," "windowing environment," and "running in windows" are used interchangeably to denote a computer user interface in which information is manipulated and displayed within bounded regions on a raster scanned video display.

The terms "application," "computer application software," and "program" are used interchangeably herein to refer to any computer program run in conjunction with the present inventive system. Such computer programs could relate to computer applications, with on-line services, communication systems, or any other computer oriented function.

The term "current" is sometimes used herein as an antecedent to "window," "application," etc., and is used to denote system components which are currently being utilized or performing operations with respect to a particular computer application software running in the environment.

A system 10 embodying the principles of the present invention is shown by way of illustration in FIGS. 1-4. The system 10 of the present invention is implemented on a typical computer system 11 as shown in FIG. 1. This computer system 11 typically comprises a CPU 12, a computer bus 14, a disc drive 16, main memory 18, a compact disc drive 20, and user interface components. These user interface components preferably comprise a mouse 22 and mouse controller 24, a video display 26 and video display controller 28, and a keyboard 30 and keyboard controller 32. Preferably, the computer system 11 also includes an audio interface 34 that transmits audio information to and receives audio information from a user of the system.

As one skilled in the art will appreciate, the system and method of the present invention are implemented on the computer system 11 but are not readily identifiable as specific components of the system. Those skilled in the art will readily understand how the described invention may be implemented on any of a variety of computer systems. Therefore, the implementation of the system on a particular hardware platform will not be more fully described herein.

Referring now to FIGS. 2 through 4, a computerized, multimedia tutorial interface system 10 for training a user to use computer application software comprises control display means 40, instruction input means 42, instruction interpretation means 44, audiovisual enablement means 46, computer application software interface means 48, and computer application software control means 50.

Referring specifically to FIGS. 2 and 3, the control display means 40 comprises a control bar 51 and a topic selection screen 53, each of which are selectively displayed on the computer screen 26. Together, the control bar 51 and the topic selection screen 53 provide a plurality of instructions to a user that may be selected by the user. As is shown, the control bar 51 preferably has commands similar to those found on a video tape player, including exit, rewind, goto, fast forward, stop, back, pause, and play. The topic selection screen 53 allows a user to access video segments relating to specific topics to be learned or specific operations to be executed. Typically, the user accesses specific video segments on a topic-by-topic basis as desired. In the preferred embodiment, the topic selection screen 53 is displayed only when certain commands are executed via the control bar 51. However, depending upon the application, the topic selection screen 53 could also be continuously displayed.

The instruction input means 42 operates to receive an instruction from a user 49. Preferably, the instruction input means 42 combines hardware and software components.

In the preferred embodiment of the present invention, the instruction input means 42 comprises the combination of the mouse 22, the keyboard 30, the control bar 51, the topic selection screen 53, and related software that allows the user 49 to select a desired function. Selecting a command from a menu displayed on a computer screen 26 using a mouse 22 and via a keyboard 30 are both well known in the art and are not fully described herein. As one skilled in the art will readily appreciate, however, the instruction input means 42 could also include the combination of the audio interface 34 in conjunction with voice recognition software.

The instruction interpretation means 44 interprets the user instruction, creates at least one execution instruction, and selectively issues the execution instruction. The instruction interpretation means 44 preferably comprises a series of software instructions executed on the computer system 11 in a fashion well known in the art. For example, when the user selects an instruction via the instruction input means 42, software code monitors the mouse controller 24 and keyboard controller 32 interfaces, receives input from the interfaces, processes the input to determine what function has been executed, and issues the proper execution instruction to the respective system component.

Still referring to FIGS. 2 and 3, the audiovisual enablement means 46 operates to receive execution instructions from the instruction input means 44, to selectively retrieve audiovisual information responsive to the execution instruction, and to display the audiovisual information on the computer screen 26. Preferably, the audiovisual information comprises a video clip that is retrieved from a compact disc via the CD drive 20 or the data link. The video clip is then decoded, formatted, and displayed on the computer screen 26 in a video window 55. The video window 55 may cover only a portion of the computer screen 26 or may be expanded to be as large as the screen. Preferably, as is shown in FIG. 3, the video window 55, the control bar 51, and the topic selection screen 53 all reside on

top of the computer application software window 57 when they are active. However, when they are inactive, they are all hidden. Further, in the preferred embodiment, the topic selection screen 53 is displayed only when certain user instructions are executed.

In a typical use of the tutorial interface system, the user 49 selects a specific video clip that corresponds to a particular topic and topic to be learned. The video clip is then retrieved and displayed on the computer screen 26 in the video window 55. After the information has been displayed, and, if the process has not been aborted or otherwise interrupted by the user, control may be returned to the user or be given to another system 10 component. In the preferred embodiment, immediately after the video clip has been displayed, or during a user 49 initiated break in the video clip, control is taken again by the computer application software interface means 48. However, the system 10 may also be operated such that the computer application software interface means 48 takes control during a video clip, halts the video clip to demonstrate a function or service feature, and then later restarts the video clip.

The computer application software interface means 48 also receives execution instructions from the instruction interpretation means 44. The computer application software interface means 48 interfaces directly with computer application software and selectively executes a function of the computer application software that is described in a video clip. Thus, the functions performed by the computer application software interface means 48 within the computer application software provides a second visual training tool to the user 49 on the computer display 26. Preferably, the function or set of functions executed within the computer application software relate directly to the audiovisual segment that was just displayed to the user 49. Preferably, the computer application software interface means 48 comprises a communication agent that is loaded into main memory at system 10 startup. The communication agent accesses instruction sets specific to the

computer application software of interest that are stored in separate files on the disc drive 16. Thus, to perform a specific set of instructions within the computer application software, the communication agent brings the computer application software up on the computer display 26, accesses the instructions, and then executes the instructions within the computer application program.

The computer application software control means 50 selectively relinquishes control of the computer application software to the user 49 so that the user may practice operating the computer application software. The computer application software control means 50 also selectively regains control of the computer application software from the user 59. In this fashion, the user may practice those techniques that were previously described to him via the video clip and also were performed by the computer application software interface means 48. As one skilled in the art will readily appreciate, the computer application software control means 50 is preferably implemented as a combination of software instructions.

Preferably, the system 10 of the present invention also comprises user instruction monitoring means 52, error message issuance means 54, and evaluation means 56, all of which provide feedback to the user when the user has control of the computer application software. Specifically, the instruction monitoring means 52 monitors the user instructions issued to the computer application software, keeping track of the instructions. When activated, the error message issuance means 54 issues an error message to the user on the computer screen 26 if the user issues instructions that are erroneous. To determine whether the issued instructions are erroneous, the error message issuance means 54 compares the user's issued instructions to a list of correct instructions. Further, when activated, the evaluation means 56 evaluates the instructions issued to the computer application software by the user and issues a summary of the user's performance in issuing the instructions.

Thereby, the evaluation means 56 provides an indication of the user's performance in learning to use the computer application software. As one skilled in the art will readily appreciate, the user instruction monitoring means 52, error message issuance means 54, and evaluation means 56 are all preferably implemented as a combination of software instructions and executed accordingly.

Referring specifically to FIG. 4, the operation of the tutorial interface system 10 is described. In the description of the system 10 operation, each relevant system event is identified with a numeral in parentheses. Immediately after the system 10 is started (100), the interface with the computer application software is initiated, and the control bar 51 and video window 55 are created (102). Next, the interface between the main program and the communication agent is established (104), and the communication agent is loaded into main memory 18. At this point, the product logo is displayed, and an introduction video segment may be played (108) on the computer screen 26. The system 10 then prompts the user to enter an instruction from the control bar 51. Immediately upon entering the program, a topic index is set to a predetermined value and a topic index is also set to a predetermined value. When the program is run for the first time, these two indexes are set at one. However, when the user 49 continues with a previously-started topic, the indexes may be automatically set to those of the prior session.

Each instruction available on the control bar 51 may be executed by the user 49. The EXIT instruction (110) provides notification of an exit to the communication agent via the communication agent (112), stops the communication agent (114), and closes all communication agent command files (116). The EXIT instruction (110) optionally further shuts down the core program (118), closes the control bar 51 window, the video window 55 (120), and ends the current execution or training session (122).

Executing the REWIND (RWD) instruction (124) with a double click causes the current topic index and the current topic index, as displayed in the topic selection screen window 53, to index to the first topic and first topic (126). Executing the REWIND (RWD) instruction with a single click causes the current topic index to decrement by a single topic (126).

Executing the GOTO instruction (128) opens the topic selection screen 53 and allows the user to select a particular topic to be indexed (130). Then the user 49 has the option of playing the video clip or the demonstration of the particular topic. Depending upon the option selected, the video clip plays or the execution plays (130).

Executing the FAST FORWARD (FF) instruction (132) with a double click causes the current topic index and the current topic index, as displayed in the topic selection screen window 53, to index to the last topic (134). Executing the FAST FORWARD (FF) instruction (132) with a single click causes the current topic index to increment by a single topic (134).

Executing the STOP instruction (136) causes the process that is running when the instruction is executed to stop immediately (138). Executing the BACK instruction (140) freezes the current topic indexes and plays the previously-viewed video clip again (142). The PAUSE instruction (144) causes the currently-playing video clip, if one is playing, to stop for later continuation (146).

The PLAY instruction (148) first causes the topic counter to increment (150). The topic counter indexes the relevant video clip and communication agent instructions. The next video clip is then played in the video window 55 (152). Then, the control bar 51 and the video window 55 are hidden, and the communication agent instructions may be executed (154). A notification of play is transmitted to the communication agent (156), the order is received by the communication agent (158), and instructions are read from the

communication agent library and executed within the computer application software (160). Once the instructions are completed, a backwards notification is sent (162) so that the control bar 51 and the video window 55 are again displayed (164). Next, a short, written synopsis of the demonstration that was executed is displayed to the user 49 on the computer screen 26 (168). Then, the computer application software interface means 48 allows the user to practice within the computer application software to enforce what he or she has learned (168). Then, the user's performance is evaluated (170).

The COUNTER INFO instruction (172) toggles the time information displayed between elapsed time from the start of the video clip to the time remaining in the video clip (174). Optionally, the counter could also display the time since the user 49 logged on or the clock time. The counter hide instruction (176) toggles the counter display between being hidden or displayed (178).

The system 10 of the present invention can be easily implemented with application programs, on-line services, or any computer application software. The system 10 is generic and provides a familiar training interface that can be used in many varied situations. As one skilled in the art will readily appreciate, the system 10 of the present invention is readily transportable to provide tutorial instruction in any computer based system.

Description of a Further Embodiment

A further aspect of the present invention has an engine component for accessing local or remote instruction file libraries and for controlling peripheral devices.

Referring to FIG. 5, a general action flow 500 of the system 10 is shown. The action flow has two portions, the first being the engine interaction portion 501 and the results area 503.

The engine interaction portion 501 has four actions: a user instruction action 502 where selections are received from the user, an instruction input action 504 where instructions are retrieved, and an instruction interpreter action 508 where the instructions are executed or translated into use. Application selector action 506 is to ensure coverage of the aspect of the engine 501 being a "juke box" for instruction files, so that the engine can externally operate a wide variety of target applications. The term "externally" means that the control or operation activity comes from without the target application. In contrast, for example, an internal control or operation is commonly provided by a function wizard such as those that accompany Microsoft® Windows or Excel.

Upon start-up of the action flow, the user instruction action block (502) is provided by the control bar 51 (shown in Fig. 3). At user instruction action block (502), the choices of either a target application (506) or a tutorial program is selected from topic selection screen 53 (shown in Fig. 3) because an instruction program typically has a plurality of instruction files that can be executed randomly. That is, one of the possible user instructions is the choice of which "clip" or inserted file to play through the control bar 51 (shown in Fig. 3). Other available user instructions are user preferences for running the program such as speed and screen display options. Afterwards, at instruction input action (504), engine 702 fetches instructions that are remotely or locally located. Afterwards, after the fetched instructions are loaded into memory, the engine interprets the instructions accordingly at the instruction interpreter action block (508).

The choice of application is also a form of user instruction action (502). For example, if application programs are on the computer, such as Microsoft® Works or Excel, then the engine 702 presents a variety of instructions, one of which is to start the application program. That is, user 49 selects the target application, being Microsoft® Word for example, for which may exist a dedicated set of instructions, which are disassociated from

the user instructions 502. The instruction interpreter 508 is then called, which chooses one instruction file having a set of instruction related to Microsoft® Word and then returns along flow path 511 to the user instructions action 502 to receive the user's next instruction.

When an instruction set is ready, the next user instruction, which is typically "Play," but can also be "rewind," or other instructions as set out in the discussion regarding Fig. 3 earlier herein. The action flow then retrieves (in the instruction input action (504)) and translates the instruction into an action at the instruction interpreter action 508. The engine continues to "play" until the end of the instruction file 1000 is reached (see Fig. 12), discussed later herein.

Upon interpretation of the instruction files at the instruction interpreter action 508, the result area 503 is entered through the instruction interpreter action (508). The result area 503 has an audiovisual interface action (510), and application interface action (512), and an application control action (514). That is, the appropriate media data is played accordingly. For example, if a video clip was selected, then the clip is decoded, formatted, and displayed on the computer screen 26 in video window 55 (see Fig. 3). The audiovisual interface action (510) plays multimedia information. The application interface action (512) is the interface of application program functions with the computer peripherals such as mouse, display, keyboard and the like. The application control action (514) is the result of putting the application peripherals into action. In the case of application control action (514), checking whether everything is operating and execution is suspended or stopped accordingly in the target application. That is, the system polls the target application for status checks regarding problems or the like. For example, when computer error occurs, an error message is broadcast. So control action (514) polls the program so that the user is informed accordingly.

Referring to Fig. 6, a detail of system 10 is shown with an engine 702 having interfaces for instruction/data storage 700, a user 704, target application 708, a peripheral device 710, and a host 712. It should be noted that other interfaces can be had.

Engine 702 has a plurality of software components with designated software functions. Engine 702 has an interpreter component 716, and an interpreter component 716 that routes instructions to the appropriate application or peripheral device. Interpreter 716 is in bi-directional communication with an instruction/data interface component 714, a target interface component 720, and peripheral interface component 722. Interpreter component 716 is in uni-directional communication with user interface 718.

Target interface 720 and user interface 718 are components of system 10 shown in Fig.s 1-4. The capabilities added to the engine 702 as shown in Fig. 6 are the instruction/data interface 714 and peripheral interface 722 components adding new and additional device capabilities to the system.

The instruction/data interface component 714 can be constituted of either a local location or remote network interface location. Location instruction/data interface component 714 is a set of functions that load or access files either locally or remotely. The term "locally" means a physical data storage device such as CD drive 20 or disk drive 16, shown in Fig. 1. The term "remotely" means a data link, established either through a local area network ("LAN") or a wide area network ("WAN") or the Internet as set out under conventional network standard protocols. A suitable network standard is TCP/IP ("Transport Control Protocol/Interface Program").

Referring to Fig. 8, an instruction/data storage component program flow chart is shown. After startup (801), the engine screen is displayed (see Fig. 3). Engine 702 also determines the system hardware configuration (802). After the configuration is established, the engine 702 prompts the user 49 for a choice of which instruction to execute (804).

Upon an input selection by user 49, engine 702 determines whether data files are local (806). If files are local (808), then engine 702 retrieves the designated files (820). If the data files are not local (808), then engine 702 establishes communications network connection 810. The communications network connection can be established over a conventional computer communications devices. Examples of suitable devices are a conventional modem or ISDN ("Integrated Services Digital Network") communications links using conventional software communications services. Upon establishment of the connection (810), the user's identity is verified (812). After verification, the required instruction files are downloaded to the engine 702 and either stored in local memory or executed on-the-fly. Preferably, in light of the current technology, the files are stored locally. But as technology permits, other execution schemes can be implemented with equal effectiveness.

At this point, the retrieve files instruction step (820) and the download required files step (814) converge. The next step for the engine 702 is to execute the initialization instructions included in the files that were retrieved either locally or remotely (816). Referring back to Fig. 6, the interpreter reads the initialization instructions obtained in step (816) and routes accordingly. That is, the interpreter prepares the operating environment of the target application or the target application itself for the instruction files to be effectively executed by the engine 702. For example, if the retrieved instruction files require that the main window of the target application occupy all the screen, the engine 702 will have the target application occupy all the screen. Referring back to Fig. 7, the user is prompted for input (818). The input being the options provided by the control bar 51 (Fig. 3) such as "play," "rewind" or the like.

Referring back to Fig. 6, the input provides directions for execution on either the target interface component 720 or the peripheral interface 722 depending on the instruction type. The peripheral interface 722 is the capability by which the engine 702 can issue

instructions specific to particular peripheral devices 710 responsive to external-computer signal control such as computer printers, industrial robotics, robotic drones or the like. Conventional peripheral devices 710 typically have their own user-interface apart from the engine 702 interface component 718.

Referring to Fig. 8, the program waits for user input (822). Upon the user input, the engine 702 identifies the instructions necessary in response to the user 49 input from the control bar 51 (826). Next, the engine 702 determines the application target of the user's instruction (828). Based upon the target determination (702), the engine will know if the instructions require translation in that, if the instruction is for a target application (830), the instruction will be executed directly (834). If the instruction is for a peripheral device (830), then the instruction will be translated (832) and changed in its contents in order for it to be issued to the peripheral device in the device's communication medium (832). Then the instruction is issued and executed to the appropriate application 708 or device 710 (834).

Feedback is issued from the application 708 or device 710 (834) to the user 49 through the screen 26 shown in Fig. 1 (828). The user at any point has the ability to terminate the process. If the user 49 does not elect to terminate the process, the process will continue until the end of the retrieved instruction file is reached and that all instructions have been executed or an error has occurred that led the engine 702 to stop execution by itself.

Referring to Fig. 9, shown is an instruction file 1000 format having a sequence filed for each command. For example, each instruction in an instruction file typically has four parameters: "time stamp," "type of instruction," "parameter 1," and "parameter 2." The fields are not separated, and each instruction is not separated from the previous or the following ones. On each instruction file, there is typically a header block containing parameters relating to the environment needed for the instruction about to be executed. For example, if a target application needs to occupy all the screen 26, the header block dictates

that the sequence of instructions requires this environment. This header block is executed in the initialization instructions in step 816 of Fig. 8.

Referring to Fig. 10, an example of a library field structure that can be available to the system 10 is shown. An advantage to the system 10, which implements the engine 702 (shown in Fig. 7), is that remote access to industrial machinery in semiconductor clean rooms or in hazardous areas can be readily had with equipment implementing computer communications interfaces, for example, the RS-232-C standard for serial communications interfaces or the V.32 recommendation standard for 9600-bps modems. System link 910 is the link between instruction/data storage 700 and instruction/data interface 714. That is, system link 910 is the hardware connection to libraries 902 through 914, which can be located remotely on the Internet, LAN or, WAN systems. Referring to the clean-room environment, the engine 702 can access instruction files in these libraries that have processes for controlling semiconductor fabrication equipment remotely. Otherwise, when such equipment fails, the plant must be closed down while engineers, specially outfitted in particle-free suits, repair the clean-room equipment. A loss in production and profit occurs during the fabrication plant shut-down. The universal capability of using a general personal computer to remotely operate industrial equipment is greatly desired.

The above-described preferred embodiments are intended to illustrate the principles of the invention, but not to limit the scope of the invention. Various other embodiments and modifications to these preferred embodiments may be made by those skilled in the art without departing from the scope of the following claims.

In view of the above-detailed description of the present invention and associated drawings, other modifications and variations will now become apparent to those skilled in the art. It should also be apparent that such other modifications and variations may be

effected without departing from the spirit and scope of the present invention as set forth in the claims which follow.